Anglo-Chinese School (Barker Road)

SECONDARY FOUR EXPRESS

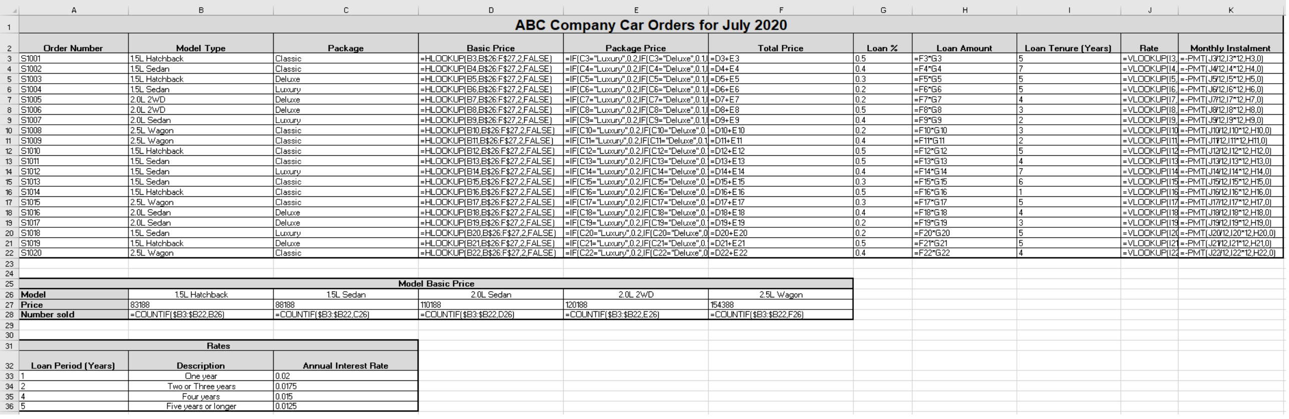
COMPUTING

PRELIMINARY EXAMINATION

Solution Paper 2

Task 1

ORDERS_sample - Formula view



	A	B	C	D	E	F	G	H	I	J	K
1	ABC Company Car Orders for July 2020										
2	Order Number	Model Type	Package	Basic Price	Package Price	Total Price	Loan %	Loan Amount	Loan Tenure (Years)	Rate	Monthly Instalment
3	S1001	1.5L Hatchback	Classic	=HLOOKUP(B3,B$26:F$27,2,FALSE)	=IF(C3="Luxury",0.2,IF(C3="Deluxe",0.1,I	=D3+E3	0.5	=F3*G3	5	=VLOOKUP(I3,	=-PMT(J3/12,I3*12,H3,0)
4	S1002	1.5L Sedan	Classic	=HLOOKUP(B4,B$26:F$27,2,FALSE)	=IF(C4="Luxury",0.2,IF(C4="Deluxe",0.1,I	=D4+E4	0.4	=F4*G4	7	=VLOOKUP(I4,	=-PMT(J4/12,I4*12,H4,0)
5	S1003	1.5L Hatchback	Deluxe	=HLOOKUP(B5,B$26:F$27,2,FALSE)	=IF(C5="Luxury",0.2,IF(C5="Deluxe",0.1,I	=D5+E5	0.3	=F5*G5	5	=VLOOKUP(I5,	=-PMT(J5/12,I5*12,H5,0)
6	S1004	1.5L Sedan	Luxury	=HLOOKUP(B6,B$26:F$27,2,FALSE)	=IF(C6="Luxury",0.2,IF(C6="Deluxe",0.1,I	=D6+E6	0.2	=F6*G6	5	=VLOOKUP(I6,	=-PMT(J6/12,I6*12,H6,0)
7	S1005	2.0L 2WD	Deluxe	=HLOOKUP(B7,B$26:F$27,2,FALSE)	=IF(C7="Luxury",0.2,IF(C7="Deluxe",0.1,I	=D7+E7	0.2	=F7*G7	4	=VLOOKUP(I7,	=-PMT(J7/12,I7*12,H7,0)
8	S1006	2.0L 2WD	Deluxe	=HLOOKUP(B8,B$26:F$27,2,FALSE)	=IF(C8="Luxury",0.2,IF(C8="Deluxe",0.1,I	=D8+E8	0.5	=F8*G8	3	=VLOOKUP(I8,	=-PMT(J8/12,I8*12,H8,0)
9	S1007	2.0L Sedan	Luxury	=HLOOKUP(B9,B$26:F$27,2,FALSE)	=IF(C9="Luxury",0.2,IF(C9="Deluxe",0.1,I	=D9+E9	0.4	=F9*G9	2	=VLOOKUP(I9,	=-PMT(J9/12,I9*12,H9,0)
10	S1008	2.5L Wagon	Classic	=HLOOKUP(B10,B$26:F$27,2,FALSE)	=IF(C10="Luxury",0.2,IF(C10="Deluxe",0.1	=D10+E10	0.2	=F10*G10	3	=VLOOKUP(I10	=-PMT(J10/12,I10*12,H10,0)
11	S1009	2.5L Wagon	Classic	=HLOOKUP(B11,B$26:F$27,2,FALSE)	=IF(C11="Luxury",0.2,IF(C11="Deluxe",0.1	=D11+E11	0.4	=F11*G11	2	=VLOOKUP(I11,	=-PMT(J11/12,I11*12,H11,0)
12	S1010	1.5L Hatchback	Classic	=HLOOKUP(B12,B$26:F$27,2,FALSE)	=IF(C12="Luxury",0.2,IF(C12="Deluxe",0.1	=D12+E12	0.5	=F12*G12	5	=VLOOKUP(I12	=-PMT(J12/12,I12*12,H12,0)
13	S1011	1.5L Sedan	Classic	=HLOOKUP(B13,B$26:F$27,2,FALSE)	=IF(C13="Luxury",0.2,IF(C13="Deluxe",0.1	=D13+E13	0.5	=F13*G13	4	=VLOOKUP(I13	=-PMT(J13/12,I13*12,H13,0)
14	S1012	1.5L Sedan	Luxury	=HLOOKUP(B14,B$26:F$27,2,FALSE)	=IF(C14="Luxury",0.2,IF(C14="Deluxe",0.1	=D14+E14	0.4	=F14*G14	7	=VLOOKUP(I14	=-PMT(J14/12,I14*12,H14,0)
15	S1013	1.5L Sedan	Classic	=HLOOKUP(B15,B$26:F$27,2,FALSE)	=IF(C15="Luxury",0.2,IF(C15="Deluxe",0.1	=D15+E15	0.3	=F15*G15	6	=VLOOKUP(I15	=-PMT(J15/12,I15*12,H15,0)
16	S1014	1.5L Hatchback	Classic	=HLOOKUP(B16,B$26:F$27,2,FALSE)	=IF(C16="Luxury",0.2,IF(C16="Deluxe",0.1	=D16+E16	0.5	=F16*G16	1	=VLOOKUP(I16	=-PMT(J16/12,I16*12,H16,0)
17	S1015	2.5L Wagon	Classic	=HLOOKUP(B17,B$26:F$27,2,FALSE)	=IF(C17="Luxury",0.2,IF(C17="Deluxe",0.1	=D17+E17	0.3	=F17*G17	5	=VLOOKUP(I17	=-PMT(J17/12,I17*12,H17,0)
18	S1016	2.0L Sedan	Deluxe	=HLOOKUP(B18,B$26:F$27,2,FALSE)	=IF(C18="Luxury",0.2,IF(C18="Deluxe",0.1	=D18+E18	0.4	=F18*G18	4	=VLOOKUP(I18	=-PMT(J18/12,I18*12,H18,0)
19	S1017	2.0L Sedan	Deluxe	=HLOOKUP(B19,B$26:F$27,2,FALSE)	=IF(C19="Luxury",0.2,IF(C19="Deluxe",0.1	=D19+E19	0.2	=F19*G19	3	=VLOOKUP(I19	=-PMT(J19/12,I19*12,H19,0)
20	S1018	1.5L Sedan	Luxury	=HLOOKUP(B20,B$26:F$27,2,FALSE)	=IF(C20="Luxury",0.2,IF(C20="Deluxe",0	=D20+E20	0.2	=F20*G20	5	=VLOOKUP(I20	=-PMT(J20/12,I20*12,H20,0)
21	S1019	1.5L Hatchback	Deluxe	=HLOOKUP(B21,B$26:F$27,2,FALSE)	=IF(C21="Luxury",0.2,IF(C21="Deluxe",0.1	=D21+E21	0.5	=F21*G21	5	=VLOOKUP(I21	=-PMT(J21/12,I21*12,H21,0)
22	S1020	2.5L Wagon	Classic	=HLOOKUP(B22,B$26:F$27,2,FALSE)	=IF(C22="Luxury",0.2,IF(C22="Deluxe",0	=D22+E22	0.4	=F22*G22	4	=VLOOKUP(I22	=-PMT(J22/12,I22*12,H22,0)
23											
24											
25	Model Basic Price										
26	Model	1.5L Hatchback	1.5L Sedan	2.0L Sedan	2.0L 2WD	2.5L Wagon					
27	Price	83188	88188	110188	120188	154388					
28	Number sold	=COUNTIF($B3:$B22,B26)	=COUNTIF($B3:$B22,C26)	=COUNTIF($B3:$B22,D26)	=COUNTIF($B3:$B22,E26)	=COUNTIF($B3:$B22,F26)					
29											
30											
31	Rates										
32	Loan Period (Years)	Description	Annual Interest Rate								
33	1	One year	0.02								
34	2	Two or Three years	0.0175								
35	4	Four years	0.015								
36	5	Five years or longer	0.0125								

Question	Answer	Marks
1	**One** mark for correct working HLOOKUP with FALSE lookup (top formula), **One** mark for the rest =HLOOKUP(B3,B$26:F$27,2,FALSE)	[2]

Question	Answer	Marks
2	**One** mark for correct conditional statement. **One** mark for multiplying with Basic Price. =IF(C3="Luxury",0.2,IF(C3="Deluxe",0.1,IF(C3="Classic",0)))*D3	[2]

Question	Answer	Marks
3	**One** mark for summing up Basic Price and Package Price. =D3+E3	[1]
4	**One** mark for multiplying Loan% to get the Loan amount =F3*G3	[1]

Question	Answer	Marks

5	**One** mark for correct working VLOOKUP with TRUE lookup (top formula. **One** mark for the rest =VLOOKUP(I3,A$33:C$36,3,TRUE)	[2]

Question	Answer	Marks
6	**One** mark for working top formula. (Accept negative answers) =-PMT(J3/12,I3*12,H3,0)	[1]

Question	Answer	Marks
7	**One** mark for working top formula. =COUNTIF($B3:$B22,B26)	[1]

Task 2

MYBMI

```
students = 15
upp_bound = 25
low_bound = 18.5
underwt, overwt = 0, 0

for count in range(students):
    weight = float(input('Enter weight of student in kg '))
    height = float(input('Enter height of student in cm '))
    bmi = weight/height**2 * 10000
    if bmi > upp_bound:
        print('Student is overweight')
        overwt += 1
    elif bmi < low_bound:
        print('Student is underweight')
        underwt += 1
    else:
        print('Student\'s weight is normal')

print("Number of overweight students ", overwt)
print("Number of underweight students ", underwt)
```

MYBMI2

```
students = int(input("Enter number of students "))
upp_bound = 25
low_bound = 18.5
underwt, overwt = 0, 0

for count in range(students):
    weight = float(input('Enter weight of student in kg '))
    while weight < 30 or weight > 150:
        print('Invalid weight')
        weight = float(input('Enter weight of student in kg '))
    height = float(input('Enter height of student in cm '))
    while height < 80 or height > 200:
        print('Invalid height')
        height = float(input('Enter height of student in cm '))
    bmi = weight/height**2 * 10000
    if bmi > upp_bound:
        print('Student is overweight')
        overwt += 1
    elif bmi < low_bound:
        print('Student is underweight')
        underwt += 1
    else:
        print('Student\'s weight is normal')

print("Number of overweight students ", overwt)
print("Number of underweight students ", underwt)
```

Task 3

MYMARKS

```
nlist = ["Alden", "Belle", "Charles", "Dolly", "Elle", "Falken", "Grace",
"Hacken"]
mlist = [56, 64, 23, 78, 53, 46, 98, 33]

to_find = input("Which name would you like to search for? ")

items = len(nlist)
num = 0
name_found = False
while name_found == False:
    while num < items:
        if nlist[num] == to_find:
            print("{} score {} for the test".format(nlist[num], mlist[num]))
            name_found = True
            num = items
        elif num == items - 1:
            print("{} is not in the list".format(to_find))
            name_found = True
            num = items
        else:
            num += 1
```

Question	Answer	Marks
10	**One** mark for correct list syntax `mlist = [56, 64, 23, 78, 53, 46, 98, 33]`	[1]
	One mark for getting correct number of items `items = len(nlist)`	[1]
	One mark for correct while condition `while name_found == False:`	[1]
	One mark for correct while condition `while num < items:`	[1]
	One mark for correct syntax and **one** mark for : `if nlist[num] == to_find:`	[1] [1]
	One mark for correct index `print("{} score {} for the test".format(nlist[num], mlist[num]))`	[1]
	One mark for correct logic `num = items`	[1]
	One mark for correct logic `name_found = True`	[1]
	One mark for incrementing `num += 1`	[1]

Task 4

TPSTAFF

```
no_staff = 5

staff_list = []
time_in = []
time_out = []
work_time = []
wage = []

for i in range(no_staff):
    staff = input("Enter name of staff: ")
    staff_list += [staff]
    tin = input("Time-in HH:MM for {}: ".format(staff))
    while not (tin[:2].isdigit() and tin[2] == ":" and tin[-
2:].isdigit()\
          and len(tin) == 5 and 0 <= int(tin[:2]) <= 23 and 0 <=
int(tin[-2:]) <= 59):
        tin = input("Invalid! Time-in HH:MM for {}:
".format(staff))
    time_in += [tin]

    tout = input("Time-out HH:MM for {}: ".format(staff))
    while not (tout[:2].isdigit() and tout[2] == ":" and tout[-
2:].isdigit()\
          and 0 <= int(tout[:2]) <= 23 and 0 <= int(tout[-2:])
<= 59 and len(tout) == 5\
               and (int(tout[:2])> int(tin[:2]) or int(tout[:2])
== int(tin[:2]) and \
                    int(tout[-2:]) >= int(tin[-2:]))):
        tout = input("Invalid! Time-out HH:MM for {}:
".format(staff))
    time_out += [tout]
    work = (int(tout[:2])- int(tin[:2]))*60 + int(tout[-2:]) -
int(tin[-2:])
    work_time += [work]

print()
for i in range(no_staff):
    print("{} worked for {} minutes".format(staff_list[i],
work_time[i]))

print()
print("Average number of minutes worked:
{}".format(round(sum(work_time)/len(work_time),1)))
```

RESULT screenshot

```
Enter name of staff: Andy
Time-in HH:MM for Andy: 08:05
Time-out HH:MM for Andy: 13:55
Enter name of staff: Ben
Time-in HH:MM for Ben: 07:03
Time-out HH:MM for Ben: 07:00
Invalid! Time-out HH:MM for Ben: eight o'clock
Invalid! Time-out HH:MM for Ben: 08:00
Enter name of staff: Charles
Time-in HH:MM for Charles: 10:03
Time-out HH:MM for Charles: 13:115
Invalid! Time-out HH:MM for Charles: 13:11
Enter name of staff: Dominic
Time-in HH:MM for Dominic: 09-04
Invalid! Time-in HH:MM for Dominic: 09:04
Time-out HH:MM for Dominic: 15:35
Enter name of staff: Ethan
Time-in HH:MM for Ethan: 08:42
Time-out HH:MM for Ethan: 16:55

Andy worked for 350 minutes
Ben worked for 57 minutes
Charles worked for 188 minutes
Dominic worked for 391 minutes
Ethan worked for 493 minutes

Average number of minutes worked: 295.8
```

TPSTAFF2 (appended to TPSTAFF)

```
print()
for i in range(no_staff):
    if work_time [i] <= 240:
        wage = work_time[i]//15*3
    else:
        wage = (work_time[i]-240)//15*4 + 240//15*3
    print("{} will be paid ${}".format(staff_list[i], wage))
```

TPSTAFF3

```
staff_list = []
time_in = []
time_out = []
work_time = []
wage = []
staff = input("Enter name of staff: ")

while staff != '':
    staff_list += [staff]
    tin = input("Time-in HH:MM for {}: ".format(staff))
    while not (tin[:2].isdigit() and tin[2] == ":" and tin[-
2:].isdigit()\
          and len(tin) == 5 and 0 <= int(tin[:2]) <= 23 and 0 <=
int(tin[-2:]) <= 59):
        tin = input("Invalid! Time-in HH:MM for {}:
".format(staff))
    time_in += [tin]

    tout = input("Time-out HH:MM for {}: ".format(staff))
    while not (tout[:2].isdigit() and tout[2] == ":" and tout[-
2:].isdigit()\
          and 0 <= int(tout[:2]) <= 23 and 0 <= int(tout[-2:]) <=
59 and len(tout) == 5\
               and (int(tout[:2])> int(tin[:2]) or int(tout[:2])
== int(tin[:2]) and \
                    int(tout[-2:]) >= int(tin[-2:])))):
        tout = input("Invalid! Time-out HH:MM for {}:
".format(staff))
    time_out += [tout]
    work = (int(tout[:2])- int(tin[:2]))*60 + int(tout[-2:]) -
int(tin[-2:])
    work_time += [work]
    staff = input("Enter name of staff: ")

print()
for i in range(len(staff_list)):
    print("{} worked for {} minutes".format(staff_list[i],
work_time[i]))

print()
print("Average number of minutes worked:
{}".format(round(sum(work_time)/len(work_time),1)))

print()
for i in range(len(staff_list)):
    if work_time [i] <= 240:
        wage = work_time[i]//15*3
    else:
        wage = (work_time[i]-240)//15*4 + 240//15*3
    print("{} will be paid ${}".format(staff_list[i], wage))
```

End of Answer Key